

IN THE CLAIMS

1. (Currently amended) A ~~Data~~ data processing device, ~~at least~~ comprising a master controller (1), a first functional unit (2) ~~which includes~~ including a slave controller (20), a second functional unit (3), ~~which functional units (2,3) share and~~ a common memory means shared by the first and second functional units (1-1), the data processing device being programmed for executing an instruction by the first functional unit (2), the execution of said instruction involving input/output operations by the first functional unit (2), wherein said execution involves at least one of: output data of the first functional unit (2) is being processed by the second functional unit (3) ~~during in the midst of execution of said instruction execution,~~ and/or the input data to the first functional unit being is generated by the second functional (3) unit during in the midst of execution of said instruction execution.
2. (Currently amended) The data Data processing device according to claim 1, characterized in that wherein the first functional unit (2) is arranged for processing instructions of a first type corresponding to operations having a relatively large latency and in that the second functional unit (3) is arranged for processing instructions of a second type corresponding to operations having a relatively small latency.
3. (Currently amended) The data Data processing device according to claim 1, having halt means controllable by the master controller for suspending operation of the first functional unit (2).

4. (Currently amended) A method of operating a ~~dataprocessor~~ data processing device, which ~~device comprises at least~~ comprising:
- a master controller (1) for controlling operation of the data processing device,
 - a first functional unit (2), which includes a slave controller (20), the first functional unit (2) being arranged for executing instructions of a first type corresponding to operations having a relatively long latency,
 - a second functional unit (3) capable of executing instructions of a second type corresponding to operations having a relatively short latency, wherein the first functional unit (2) during execution of an instruction of the first type receives input data and provides output data, and said execution involves at least one of:
according to which method the output data is of the first functional unit being
processed by the second functional unit (3) during in the midst of execution of
said instruction execution, and/or the input data is to the first functional unit
being generated by the second functional unit (3) during in the midst of execution
of said instruction execution.
5. (Currently amended) The method ~~Method~~ according to claim 4, characterized in that, wherein the master controller (1) temporarily suspends operation of the first functional unit (2) during execution of instructions of the first type.
6. (Currently amended) A method for compiling a program into a sequence of instructions for operating a processing device comprising a master controller, a first functional unit including a slave controller, a second functional unit, and a

common memory means shared by the first and second functional units, the method comprising: ~~according to claim 1~~

composing a model ~~is composed~~ which is representative of the input/output operations involved in the execution of an instruction ~~instructions~~ by a the first functional unit (2),

on the basis of this model, scheduling instructions for the ~~one or more~~ second functional unit ~~units~~ (3) are scheduled for performing at least one of: providing input data for the first functional unit (2) when it the first functional unit is in the midst of executing an instruction in which said input data is used, and[/or for] retrieving output data from the first functional unit (2) when it the first functional unit is in the midst of executing an instruction in which said output data is computed.

7. (Currently amended) ~~A~~ The method according to claim 6, ~~characterised in that~~ wherein the model is a signal flow graph.